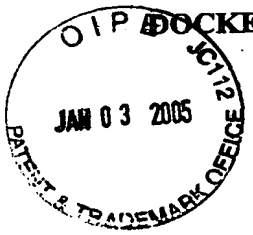
01/05/05

IDW
AF



DOCKET NO.: USYS-0046/TN128 **PATENT**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application of:
Timothy M. Young,
Application No.: 09/363,339
Filing Date: July 29, 1999
For: Voice Messaging System With Enhanced Customizability

Confirmation No.: 8021
Group Art Unit: 2645
Examiner: Simon P. Sing

EXPRESS MAIL LABEL NO: EV397436179US
DATE OF DEPOSIT: January 3, 2005

EV397436179US

MS Appeal Brief Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF REPLY BRIEF PURSUANT TO 37 CFR § 1.193

Transmitted herewith in triplicate is the REPLY BRIEF in this application with respect to the Examiner's Answer dated **November 2, 2004.**

If any fee is required, please charge Deposit Account No. 23-3050. A duplicate of this transmittal is attached.

Date: January 3, 2005

Richard W. Knight
Registration No. 42,751

Woodcock Washburn LLP
One Liberty Place - 46th Floor
Philadelphia PA 19103
Telephone: (215) 568-3100
Facsimile: (215) 568-3439

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:
Timothy M. Young, Steven J. Capriotti, Steven Luzeski, Barbara E. Osder **Confirmation No.: 8021**

Application No.: 09/363,339 **Group Art Unit: 2465**

Filing Date: July 29, 1999 **Examiner: Simon P. Sing**

For: VOICE MESSAGING SYSTEM WITH ENHANCED CUSTOMIZABILITY

EXPRESS MAIL LABEL NO: EV397436179US
DATE OF DEPOSIT: January 3, 2005

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

APPELLANT'S REPLY BRIEF PURSUANT TO 37 C.F.R. § 1.193

Appellant submits this Reply in response to the Examiner's Answer and Advisory Action dated November 2, 2004 ("Examiner's Answer") in connection with the above-identified application. As discussed in greater detail below, Appellants respectfully submit that the outstanding rejections of the pending claims as allegedly being anticipated and/or obvious are improper and should be withdrawn, and Appellants therefore respectfully request that the final rejection be reversed and that the application be remanded to the examining group for immediate allowance.

I. Detailed Reply to Examiner's Continued Rejections

A. Claims 1, 2, 8-15, 17, 18, and 20-24 are not anticipated by Sattar.

Claims 1, 2, 8-15, 17, 18, and 20-24 were rejected under 35 U.S.C. § 102(b) as being anticipated by Sattar et al. (U.S. Patent No. 5,243,643). Appellants respectfully submit that Claims 1, 2, 8-15, 17, 18, and 20-24 are not anticipated by Sattar for the reasons set forth in the Appeal Brief filed August 5, 2004 ("Appeal Brief") and the reasons set forth herein below.

Preliminarily, it should be noted and appreciated that Appellants have defined certain terms in the present Application to distinguish these terms from other generalized and often-confused definitions and understandings of the same terms in order to adequately and fully describe and disclosure the invention. It is not uncommon to have a word mean different things in different contexts. A boot can be a trunk of a car, a kind of a shoe, a process by which a computer starts up, or something to equalize an exchange (as used in tax law), for one example. Appellants respectfully submit that the Examiner's presentation has mischaracterized the words of the claims by ignoring their use in the specification in order to use similar terminology in the Sattar reference that has quite different meaning in context. This mischaracterization was used to conflate the description of the present invention with the invention of Sattar . As a result, the Examiner has inaccurately characterized the invention of Sattar using incorrectly applied words of the present Application.

In the Examiner's Answer, and in response to the Appellants' arguments, the Examiner states the following:

> Regarding claims 1 and 17, Sattar discloses a voice processing system with configurable caller interfaces. Sattar teaches a caller interface module comprising **call**

> **flows functions** (figure 2B; column 13, line 26 to column 14, lines 10), such as recording a voice message, playing a voice message, or prompts to a caller (column 9, lines 52-61). The **caller interface module** inherently includes **computer codes** (every computer program stored in a computer readable medium is coded), and a **customization list** (all the vectors listed in Appendix A reads on the claimed list), wherein the list comprises **a table** (each vector has its collection of "Events" and "Vectors" and such collection reads on the claimed tables. For example, the first vector POgRecIn in the Appendix A listed 6 events with 6 vectors for each event to perform a corresponding action) (column 28, lines 24-29).

(Examiner's Answer, paragraph 10.2) (emphasis added). To support this position, the Examiner has specifically cited Fig 2B of Sattar as well as the following text:

> Referring to FIG. 2B, an exemplary **flow diagram of state vector operation** in a telecommunications systems provided in accordance with the present invention is shown. The particular exemplary embodiment shown in **FIG. 2B integrates a number of voice processing functions**, wherein circles represent vectors, rectangles represent objects, and triangles represent particular events output by the system.
>
> As the user accesses the system, a "welcome" object 260 is first reached which provides a welcoming recording heard by the user when the system picks up. **A "play" vector 270 acts on the welcome object to change its state, thereby producing an "OK" event 280 which outputs the voice transaction found in the welcome message.**
>
> ...
>
> In this example, **the OK event 320 causes play vector 270 to change the state of the welcome object 260 which then further outputs another OK event 330. At this point, another vector 340 plays a message to the user.** After the user hears the particular voice transaction message, the system then waits to determine whether a T/O event 350 occurs, that is, the user has not taken any action to access another vector. ...

> However, if the user does not allow a T/O event to occur and wishes to record a message in the system, **a record vector 390** is accessed, producing an event, "1" at 400 indicating to the system that a message will be recorded by the user. The system then accesses **an "end" vector 410** to output an ending signal to the user and to instruct the user to hang up the handset. **Alternatively, other vectors are accessible by the user after recording a message, the other vectors 420 producing yet other event-based voice transactions which are recognizable to the user.** ...

(Sattar, col. 13, line 26 through col. 14, line 10).

Using the terminology of the art and the current Application let us put this reference to Sattar in context First, note that the basic elements of a telephony messaging application or VMA include **call flows (a.k.a., "call flow functions" changed to accommodate the Examiner's desire for clarity), functions (a.k.a., "code")**, and **prompts**. **Call flows** define the logical flow of functions in the messaging application, and are often maintained in an **interpretive state table**. **Functions** (or code) are defined in these arts as the software building blocks used to implement the desired call flow functionality and, as such, comprise **executable object code**. **Prompts** are defined in these arts as logical groups or sequences of **pre-recorded voice messages** that often reside in a database and which are utilized to direct a caller through control elements of various call flows. (Application, page 7, lines 1-17.) The combination of a **call flow** and a **function** together to perform a specific task is referred to as a **"module."** (Application, page 14, lines 3-4). Thus the Appellants' definitions are consistent with those of the art, and extend it to cover the description of the new invention by the Applicant.

In light of the foregoing, Appellants respectfully submit that what the Examiner has erroneously referred to as "call flow functions" (that do things such as "recording a voice

message, playing a voice message, or prompts to a caller") are in fact explicitly termed "vectors" in the Sattar reference (see, e.g., the circular elements of Fig. 2B), and that vectors are much more akin to "functions" than to "call flows" as those terms (and their equivalents) are used in the present application for several reasons. For example, as stated in the specification of the present application, a "function" is a "software building block" or "computer code" (executable object code) (Application, page 7, lines 6-9). Similarly, vectors are "software programs" that are preferably "programmed in C-language" and are adaptable only through changes in this C-language code (Sattar, col. 11, lines 42-64.) In contrast, "call flows" of the present Application "are often maintained in an interpretative state table" (Application, page 5, lines 2-3) which seem at most to be somewhat akin to the "exemplary flow diagram of state vector operations" as shown in Fig. 2B of the Sattar reference.

In addition, a "function" in the present Application is used to implement desired call flow functionality to "produce a result that determines the path of the associated call flow" (Application, page 7, lines 6-9). Similarly, a "vector" in Sattar performs "voice processing functions" in much the same fashion (Sattar, Fig. 2B and col. 13, line 34 to col. 14, line 10). In contrast, "call flows" of the present Application merely define the "logical flow of functions in the messaging application" just as the "exemplary flow diagram of state vector operation" of Fig. 2B "integrates a number of voice processing functions [vectors]" and, thus, merely defines the paths between the various vectors shown in Fig. 2B.

The Examiner has also separately mischaracterized the prior art in light of the present Application by equating the term "code" (functions) of the present Application with the generic term "computer codes" and a much broader definition that the Examiner characterizes by parenthetically stating that "every computer program stored in a computer readable

medium is coded" (Examiner's Answer, paragraph 10.2). However, as used in the present application, the term "code" is merely used as a synonym for "function" which, again, specifically refers to executable code for producing a result that determines the path of the associated call flow.

Last but not least, Appellants submit that the Examiner's has also micharacterized the term "Customization List" as it applies to the Sattar reference but failing to appreciate the definition provided for this term in the Specification of the present Application. Specifically, the Examiner parenthetically alleges that "all the vectors listed in Appendix A reads on the claimed list" because "each vector has its collection of "Events" and "Vectors" and such collection reads on the claimed tables" such as "the first vector POgRecIn in the Appendix A listed 6 events with 6 vectors for each event to perform a corresponding action" (Examiner's Answer, paragraph 10.2). In other words, the Examiner refers to the DTMF mappings stored in each individual vector (which, again, is equivalent to a function in the present Application) as being equivalent to a Customization List as that term is defined in the present Application. However, as set forth in the present Application, the utilization of a "Customization List comprising a table with a list of names and a modifiable list of corresponding DTMF signal identifies" must, by definition, exist distinctly and separately from the functions as illustrated in Fig. 4 of the present Application as the functions are compiled code while the Customization list is not—a position clearly supported and disclosed by Fig. 4. By existing separately from the compiled code of the functions, a customer is able to modify the DTMF mapping without having to changing the call flows and thus, unlike the "tables" and "DTMF signal" hardcoded in the vectors of the Sattar invention, the Customization Lists of the present invention are distinct and separate from the functions of the present invention. Significantly, Sattar lacks any kind of customization list that is both separate and

distinguishable from the vectors and not hard-coded directly into the vectors (what the Examiner would call "functions") in accordance with the prior art as claimed in the present Application (see Application, page 19, lines 16-18). Moreover, it is Appellants' contention that only by impermissibly viewing the Appellants' disclosure in hindsight could a person of skill in the art read into the Sattar reference the functionality and benefits of the invention described and disclosed in the present Application, and that absent the present Application the Sattar reference cannot be found to teach a system that in any way uses the concept of DTMF signals mapped to modules in a Customization List that is distinct and separate from the functions (vectors). Without such mapping of DTMF signals to modules in a Customization List, one cannot practice the Appellants' invention, and Sattar ***cannot*** provide for the ease of use in switching the mapping of functions to DTMF signals or DTMF signal groups that Appellants' invention provides.

In summary, Appellants respectfully submit that, in the invention of Sattar, the DTMF signal mapping is stored in the functions (vectors) such as the DTMF signal mapping stored in the software listing of Appendix A thereof (*see also* Sattar, col. 28, lines 6-10 and 18-24). Consequently, Sattar lacks any kind of customization list that is both separate and distinguishable from the vectors (code and call flows) and not hard-coded directly into the vectors in accordance with the prior art as claimed in the present Application (*see* Application, page 19, lines 16-18). For these reasons, Appellants respectfully submit that the utilization of a customization list as claimed in the present Application is patentably distinguishable from the inventions disclosed in the Sattar references, and Appellants seek relief from the Examiner's incorrect conclusions regarding the teachings of the prior art.

B. Claims 3-7, 16, 19, and 25 are not obvious over the combination of Sattar with other references.

Claims 3-5 and 19 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Sattar in view of Matthews et al. (U.S. Patent No. 4,652,700). Claims 6 and 7 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Sattar in view of Matthews and further in view of Weber (U.S. Patent No. 6,094,239). Claims 16 and 25 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Sattar in view of Chencinski et al. (U.S. Patent No. 5,355,406). However, with regard to these claims, none of Matthews, Weber, or Chencinski have been cited by the Examiner as disclosing the utilization of a modifiable Customization List that is distinct and separate from the executable-code functions, and thus none of these references overcome the shortcomings of the Sattar reference discussed earlier herein. Therefore, Applicants respectfully submit that Claims 3-7, 16, 19, and 25 are allowable for the same reasons given for the allowability of Claims 1, 2, 8-15, 17, 18, and 20-24 presented herein above.

II. Appellants' Reply to Examiner's "Response to Arguments"

To further clarify the differences between the Examiner's allegations and the Appellents' arguments presented herein, Appellants respectfully submit a point-by-point analysis of the Examiner's key comments in the "Response to Arguments" section of the Examiner's Answer. For convenience, the Examiners comments are italicized.

1) ***"Examiner does not clear*** **[sic]** ***if Applicants want to argue that 'call flow functions' are not the same as 'functions'. If this is the case, then how can 'functions' of call***

flows be distinct from 'functions'?" **(Examiner's Answer, paragraph 11.2, subparagraph 1.)**

Appellants respectfully submit that the term "call flow functions" is synonymous with and fully equivalent to the term "call flows" as used in the present Application, and "call flows" (and thus "call flow functions") are indeed distinct and separate from "functions" (or "code"). As discussed in detail in the Appeal Brief, the term "call flow functions" was specifically introduced at the behest and suggestion of the Examiner to clarify that this particular claim element was not merely referring to an abstract concept.

Specifically, in the First Office Action, the Examiner rejected the original Claim 1 under 35 U.S.C. § 112 observing that "applicant claims a {software} module comprises or contains call flows" and alleging that "[i]t is known in the art that a software program cannot comprise a call flow itself" and that "[a] software program can only comprise the functions of a call flow" (First Office Action, page 2, section 1.1).

To further prosecution, but without conceding the appropriateness of this rejection (particularly since call flows are often maintained in an interpretive state table and thus can comprise part of a software program), the Appellants amended Claim 1 as follows (with revisions shown):

> 1. A telephony-based messaging system application stored on a computer readable medium for use by a particular customer, comprising:
>
> a module comprising call flow[s] <u>functions</u>, code and a customization list;
>
> wherein the customization list comprises a table with a list of names and a modifiable list of corresponding DTMF signal identifiers, whereby the particular customer is permitted to change the mapping between caller-entered DTMF signals and the

corresponding actions taken by the messaging system by modifying the list of DTMF signal identifiers.

This amendment to change "call flows" to "call flow functions" was specifically intended to clarify that what is claimed are call flows embodied in a software program such as in an interpretive state table described in the specification, and not just the idea or algorithm of call flows without being embodied in a tangible medium.

2) ***"Also, the Examiner likes to point out that claims 1 and 17 merely state: 'a module comprising call flow functions, code and a customization list'. The claims do not exclude the customization list relating to call flow functions."*** **(Examiner's Answer, paragraph 11.2, subparagraph 1.)**

Again, Appellants respectfully submit that the term Customization List as defined by the Specification of the present Application does indeed preclude DTMF signals hardcoded in a function. (Without being unnecessarily glib and with all due respect, the Appellants humbly submit that it is the Appellents—not the Examiner—who are entitled to be their own lexicographer with regard to defining specialized terms used in the present Application, both in the Specification and the Claims.)

3) ***"Furthermore, in the current invention, call flow functions are computer codes (stored in a standard module) as stated in the Specification, page 17, liens 26-27: 'A module for a specific customer is basically the standard module 'compiled' with the appropriate Customization List'."*** **(Examiner's Answer, paragraph 11.2, subparagraph 1.)**

Appellants respectfully submit that the Examiner has taken this section from the Specification out of context and failed to appreciated that the word "compiled," like several

other words in the Specification, is explicitly encapsulated in quotes to suggest a non-standard contextual definition. For example, the quote-encased term "mail boxes" (Specification, page 1, line 22) does not refer to U.S. Postal Service mailboxes, and the term "agent" (Specification, page 10, line 10) does not refer to a spy, even though such alternative definitions are not explicitly provided. On the other hand, certain quote-encapsulated terms in the Specification are explicitly given with definitions, e.g., "dialog" (Specification, page 8, line 29) and "clone" (Specification, page 13, line 24). With regard to the term "compiled"—which first occurs encapsulated in quotes on page 5, lines 25-26—although an explicit definition is not provided, it is unreasonable for the Examiner to automatically presume that a source-code-to-object-code transformation must be taking place when, instead, a more reasonable reading would be that that the Customization List, along with the call flows and functions, are grouped together somehow.

In contrast, and as previously discussed, a Customization List—comprising a table with a list of name and a **customer-modifiable** list of corresponding DTMF signal identifies—*must* exist distinctly and separately from the functions as illustrated in Fig. 4 of the present Application as the functions are executable object code while the Customization list is not—the separateness of the elements being clearly support and disclosed by Fig. 4 and the necessity of which enables the customer to perform the DTMF signal mapping modification (which is not possible if the DTMF signal mapping is compiled code)—that is, by existing separately from the compiled code of the functions, a customer is able to modify the DTMF mapping without having to changing the functions.

4) ***"Sattar teaches a list with functions, but the claims (claim 1 and 17) do not state that the customization list does not have functions. Further, the customization list***

of current invention is related to call flow functions. In the specification page 17, lines 26- 28: 'A module for a specific customer is basically the standard module 'compiled' with the appropriate Customization List. The output of this build or compile is a set of files that is loaded by the NAPTool Runtime Environment during application initialization'. Also in line 29: 'Customization list should also be used in the Main module: It is clear that the customization list may be separate from, or put into, a main module (which with call flow functions), and after compilation and initialization, the customization list (including the DTMF identifiers) becomes part of the call flow functions." **(Examiner's Answer, paragraph 11.2, subparagraph 2.)**

Again, Appellants respectfully submit that the Examiner has taken this section from the Specification out of context has given the Appellant-defined word "compiled" a meaning that is not present in the Specification and which is inconsistent with the Appellant-defined term "Customization List" as previously discussed.

5) ***"Claims 1 and 17 do not recite that the customization list is stored separately from the claimed call flow functions. Sattar's vectors (customization list) are related to (pointed to, or call up) call flow functions, but separated stored from the call flow functions (or subroutines) that perform the actual actions such as playing, recording, saving and deleting voice messages."*** **(Examiner's Answer, paragraph 11.3.)**

Appellants respectfully submit, as set forth in the Specification of the present Application, the term Customization List is, by definition, separate and distinct from the functions, and that no further definition is required in the claims. In contrast, Sattar explicitly

discloses a vector (function) with DTMF signals that are *not* separate from said vector, and thus the invention of the present Application as claimed is patentably distinct.

6) ***"Further, according to the Specification (page 17, lines 26-28), the customization list of the current invention is hard-coded, because it requires compilation with a standard module in order to be loaded into the (computer) application (call flow functions) for customization. A computer program which requires compilation after changes in coding, is hard-coded."*** **(Examiner's Answer, paragraph 11.4.)**

Again, Appellants respectfully submit that the Examiner has again read additional requirements into the invention of the present Application by inappropriately defining "compilation" in a manner inconsistent with the definition set forth by the Appellants in the Specification as previously discussed herein.

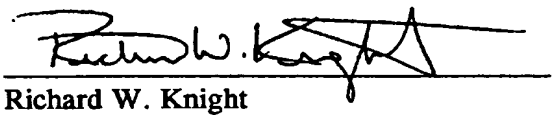
[Remainder of Page Intentionally Left Blank]

CONCLUSION

Based on the detailed arguments and analysis presented above, Appellants respectfully submit that the utilization of a customization list as claimed in the present Application is patentably distinguishable from the inventions disclosed in the Sattar reference, and Appellants seek relief from the continued reliance on incorrect conclusions regarding the teachings of this prior art based on erroneous characterizations of the meanings of the words used in the claims. More specifically, Appellants submit that the inventions recited in claims 1-25 fully comply with the requirements of 35 U.S.C. § 102 and § 103, and Appellants therefore request that this patent application be remanded to the Examiner with an instruction to both withdraw the rejections for alleged unpatentability and allow the appealed claims.

Respectfully submitted,

Date: January 3, 2005



Richard W. Knight
Registration No. 42,751

Woodcock Washburn LLP
One Liberty Place - 46th Floor
Philadelphia PA 19103
Telephone: (215) 568-3100
Facsimile: (215) 568-3439

Correspondence Address:
Unisys Corporation
Attn: Office of the General Counsel (Patents)
Unisys Way, MS/E8-114
Blue Bell, PA 19424-0001